Embedding Maps in iPhone Applications

Session 118

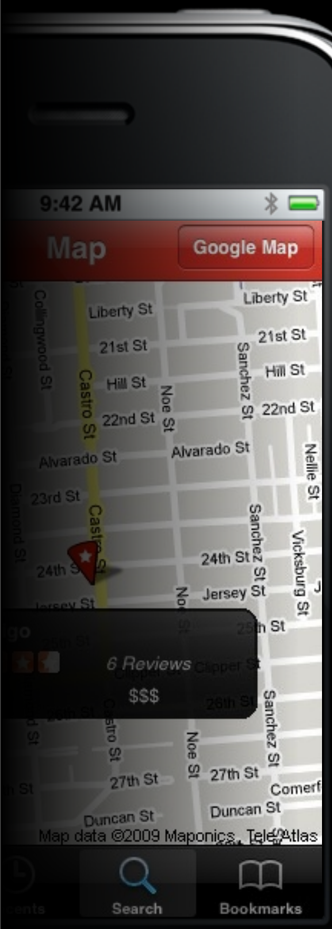
These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

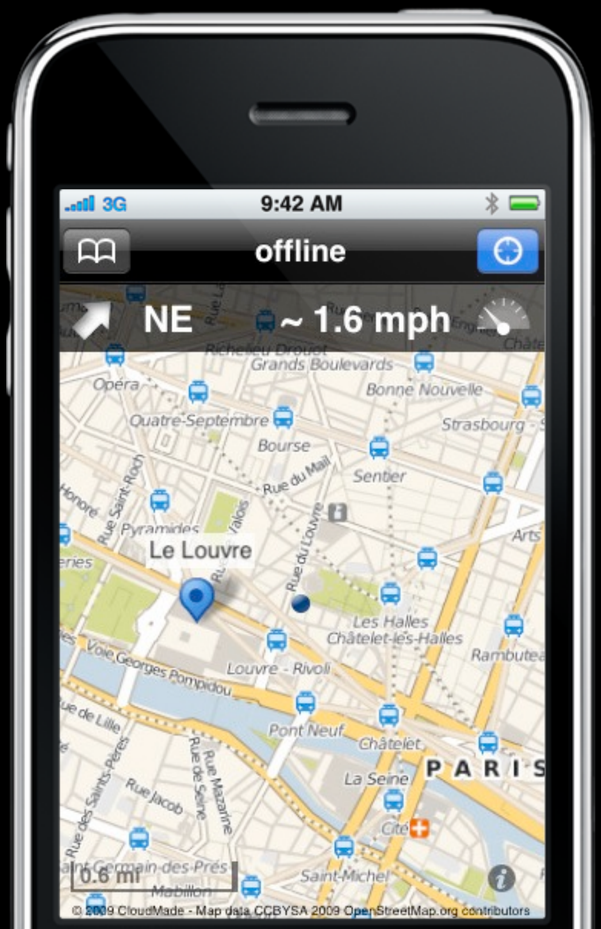
Embedding Maps in iPhone Applications

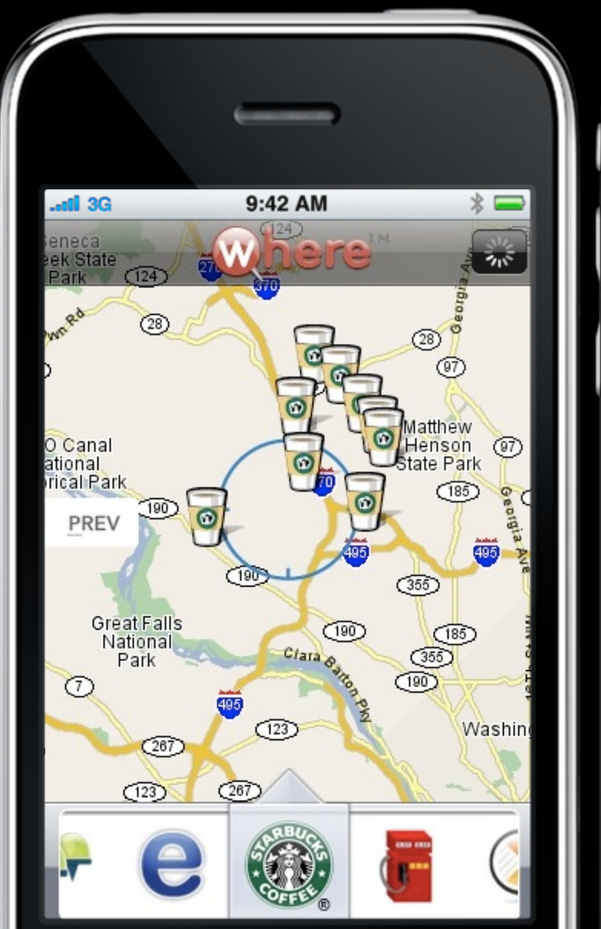
Chris Blumenberg
Manager iPhone Maps Team

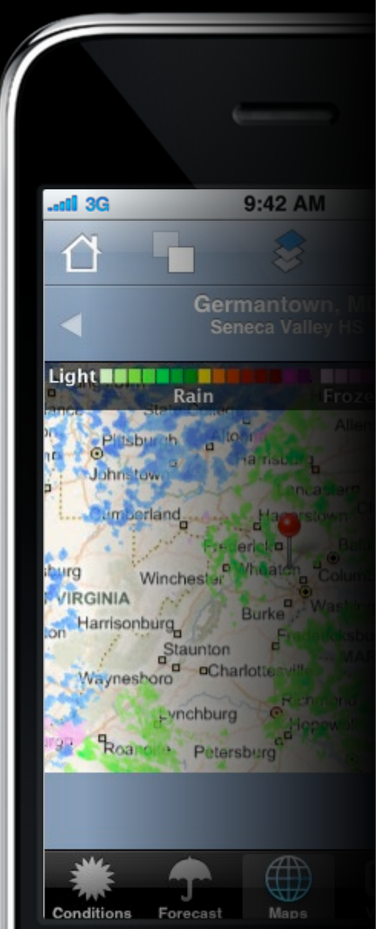
SDK
280

So Many Map Apps!









Map Kit

What Is Map Kit?

- New framework in iPhone OS 3.0
- Allows you to embed a map in your app
  - Uses Google for map data
  - Responds to panning and zooming gestures
  - You can annotate the map any way you want

What Is Map Kit?

- Reverse geocoding
  - Turns a coordinate into an address
  - Address can be turned into an Address Book contact

What You'll Learn

- Embed a map in your app using MKMapView
- Control map position and other properties
- Add view to the map
- Show the user's location in the map
- Turn a coordinate into an address using MKReverseGeocoder

MKMapView

- UIView subclass
- Works out of the box
- Loads map tiles automatically

MKMapView Performance

- Written for iPhone and iPod Touch
- Data is cached in memory and on disk
- Handles memory notifications
- Handles connectivity changes

MKMapView Map Data

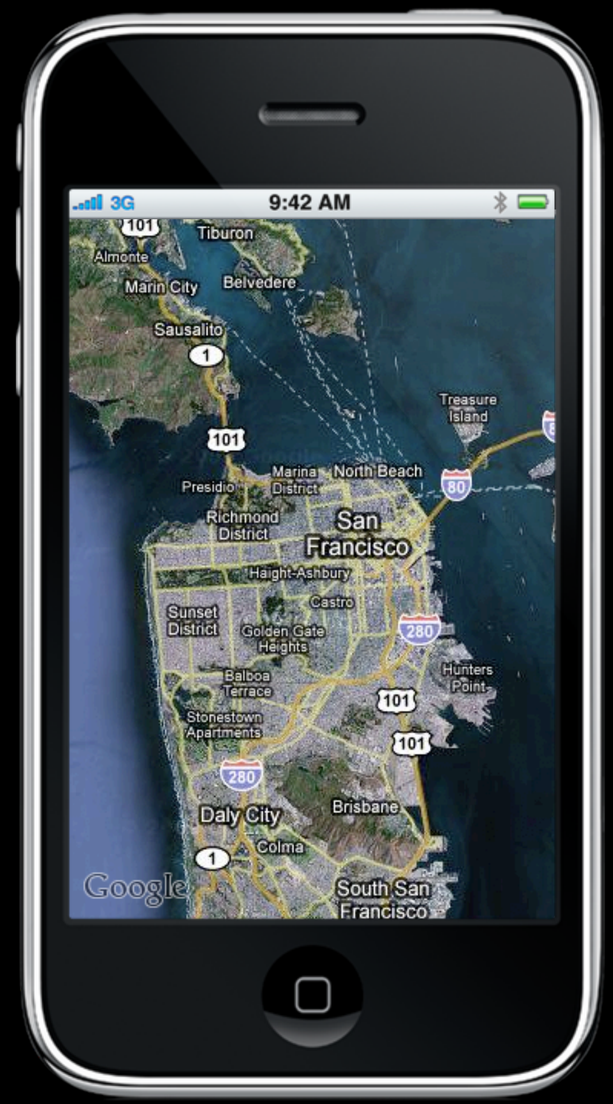
- Google hosted map data
- Network is required
- Delegate is told when map loads

MKMapView Gestures

- Scroll
- Pinch zoom
- Double-tap zoom in
- Two–finger-tap zoom out

MKMapView Map Types

- Supports map types
  - Regular
  - Satellite
  - Hybrid



MKMapView Regions

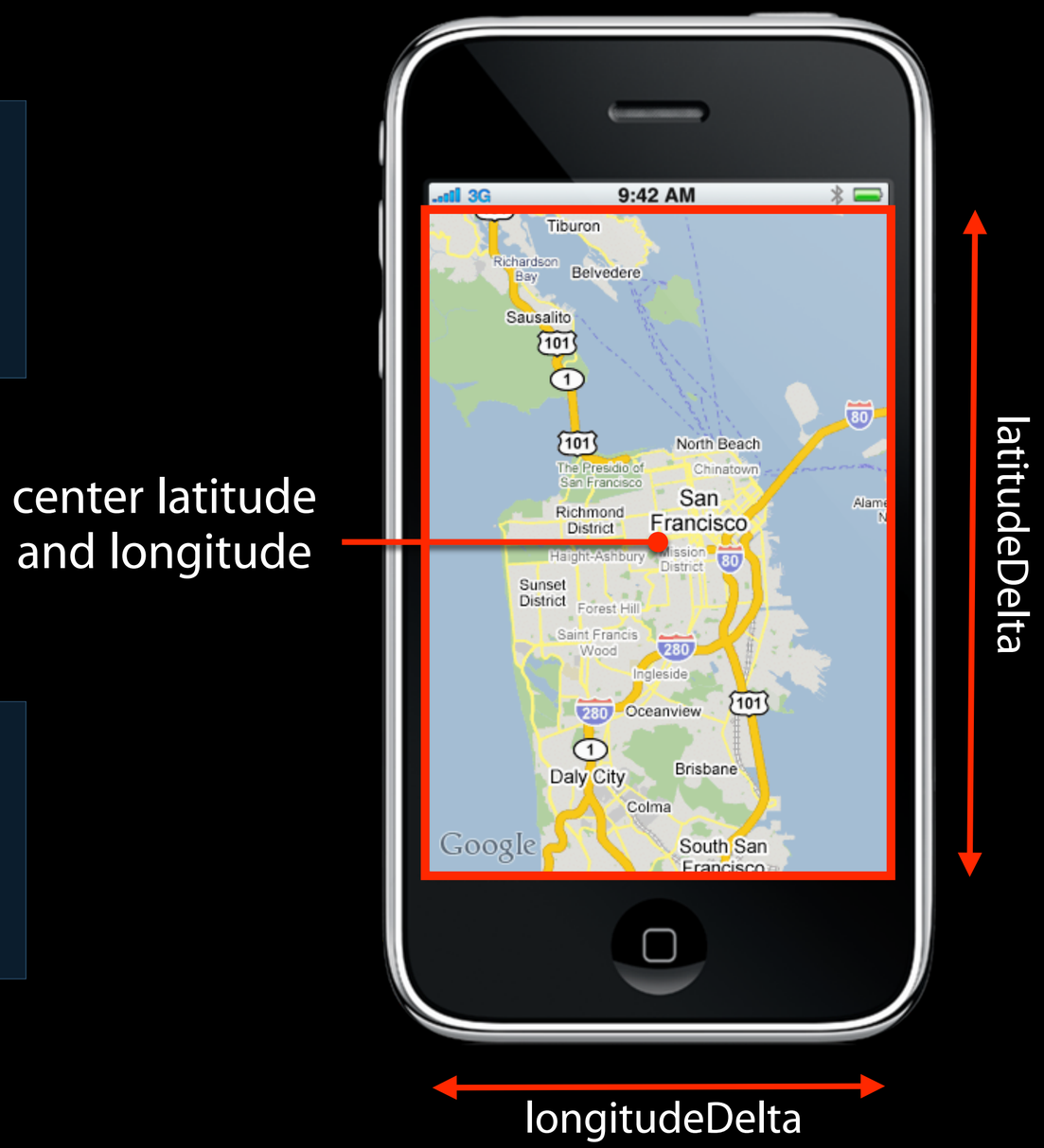
- `region` property controls map position and zoom level
- Regions are like rects
- Animatable

MKMapView Regions

```
typedef struct {
    CLLocationCoordinate2D center;
    MKCoordinateSpan span;
} MKCoordinateRegion;
```

```
typedef struct {
    CLLocationDegrees latitude;
    CLLocationDegrees longitude;
} CLLocationCoordinate2D;
```

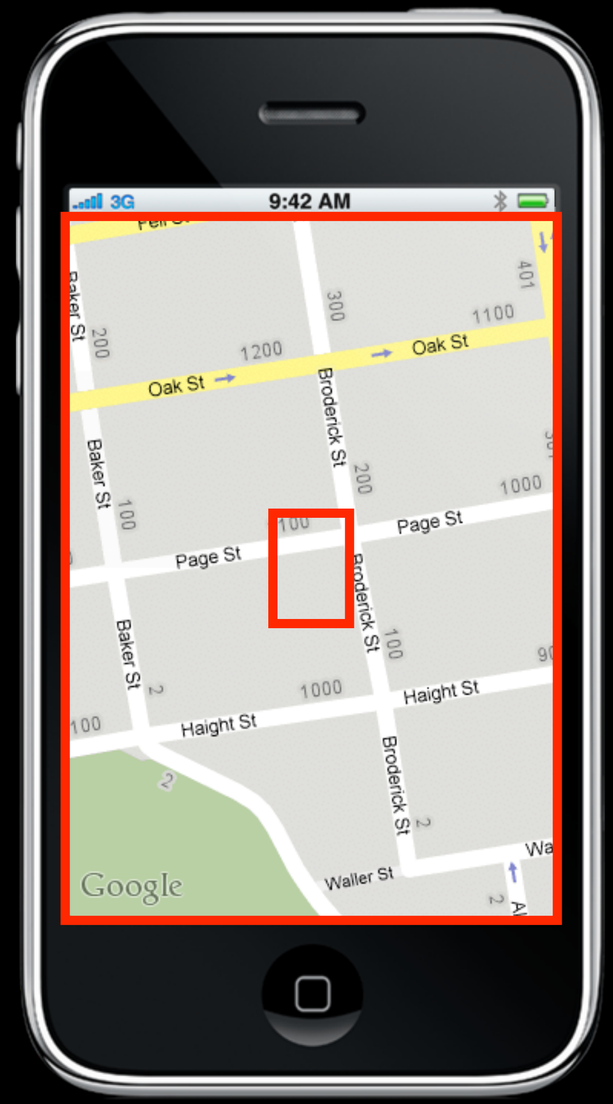
```
typedef struct {
    CLLocationDegrees latitudeDelta;
    CLLocationDegrees longitudeDelta;
} MKCoordinateSpan;
```



MKMapView Regions

- Small spans zoom in

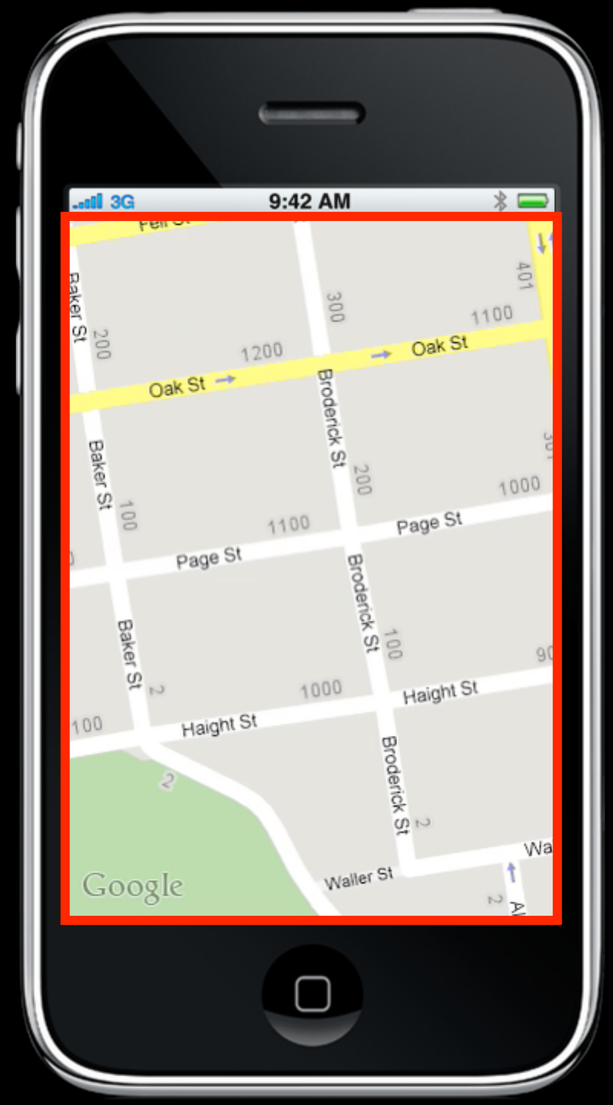
```
// Zoom into street level
MKCoordinateRegion region;

region.center.latitude = 37.7718;
region.center.longitude = -122.3492;

region.span.latitudeDelta = 0.0039;
region.span.longitudeDelta = 0.0034;

mapView.region = region;
```



MKMapView Regions

- Large spans zoom out

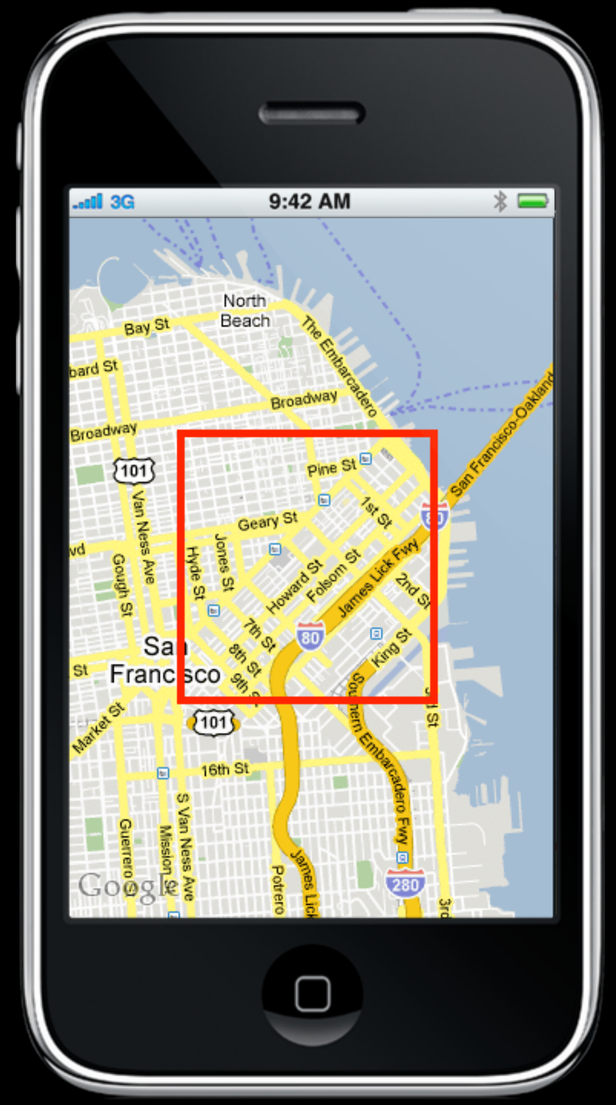
```
// Zoom out to bay area
MKCoordinateRegion region;

region.center.latitude = 37.7718;
region.center.longitude = -122.3492;

region.span.latitudeDelta = 0.9;
region.span.longitudeDelta = 0.8;

[mapView setRegion:region
          animated:YES];
```



MKMapView Regions

- Map uses zoom levels
- Map goes to the zoom level that best fits the region that is set

```
// Go to downtown SF
MKCoordinateRegion region;

region.center.latitude = 37.78338;
region.center.longitude = -122.403354;
region.span.latitudeDelta = 0.025;
region.span.longitudeDelta = 0.03;

mapView.region = region;
```

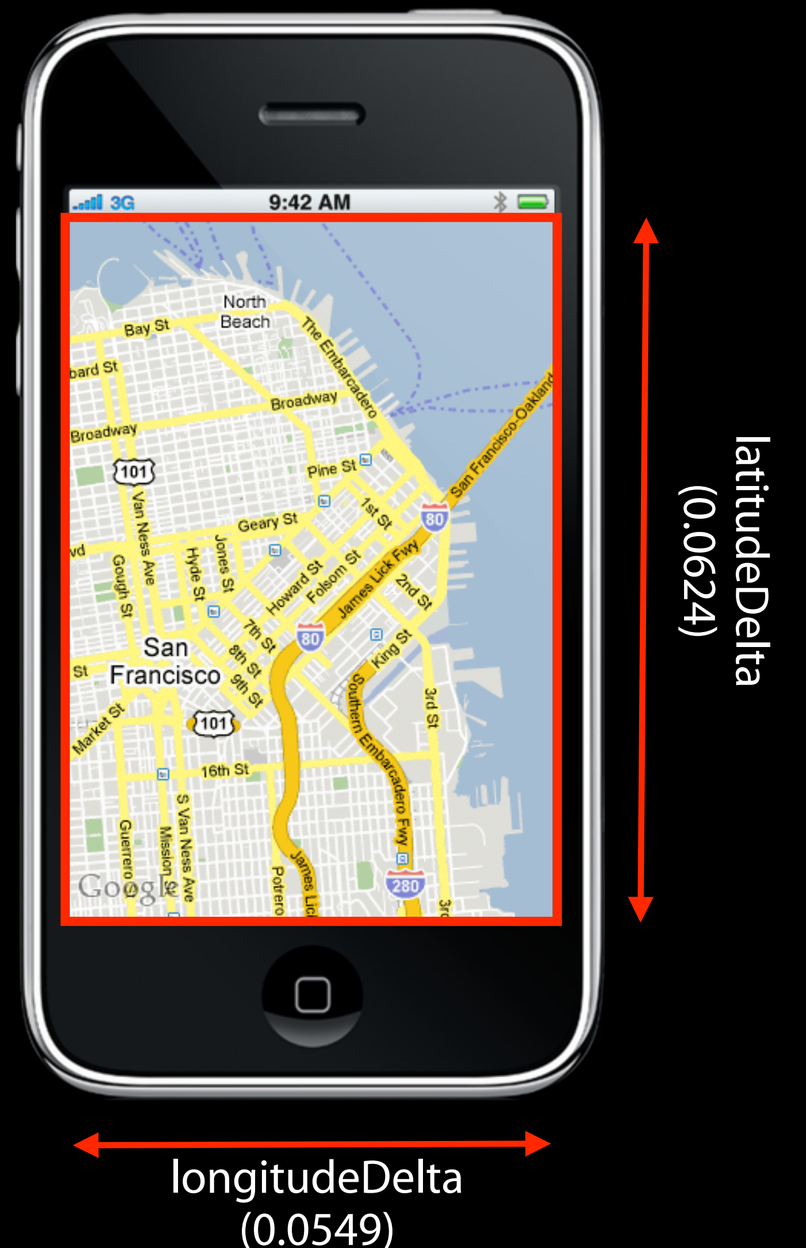


MKMapView Regions

- Resulting span will not equal the span that was set
- `regionThatFits`: Computes resulting span

```
// Go to downtown SF
MKCoordinateRegion region;

region.center.latitude = 37.78338;
region.center.longitude = -122.403354;
region.span.latitudeDelta = 0.025;
region.span.longitudeDelta = 0.03;

mapView.region = region;
```

≠



MKMapView centerCoordinate

- `centerCoordinate` property controls the map without changing the zoom level
- Animatable

Demo

Kimon Tsinteris
Software Engineer

Annotations

Annotations

- Annotate MKMapView with views
- Custom and built-in views are supported
- Annotations correspond to coordinates on the map

Annotations

- Please use annotations
- Do not add subviews directly to MKMapView
- View hierarchy within MKMapView is private

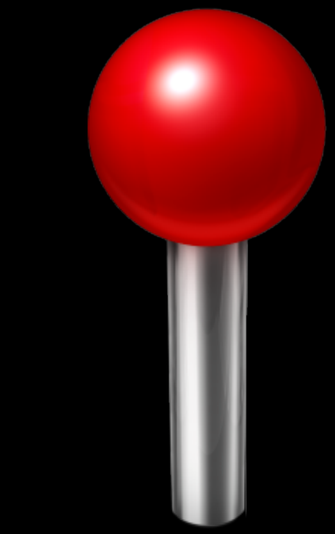
Annotations

- App provides the model and the view for each annotation
- MKMapView determines when annotation views should be created
- For large data sets, add subsets of annotations on region changes

MKAnnotation

- It's a protocol
- Your model objects conform to MKAnnotation
- App adds and removes them to/from MKMapView
- Each object must have a coordinate

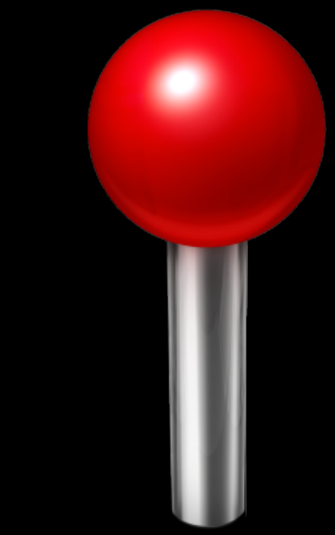
MKAnnotationView



Annotation View

- It’s a UIView subclass
- MKAnnotationView can be subclassed or used as is
- Built-in view available as well
- Like UITableViewCell, views are reusable

MKAnnotationView



Annotation View

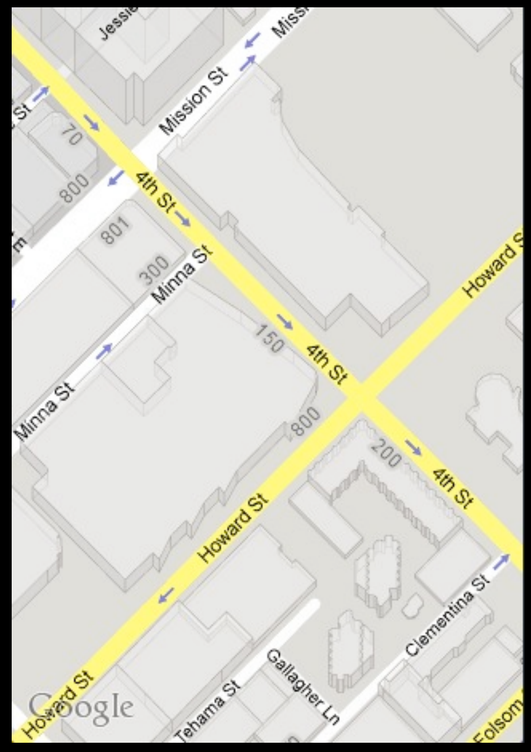
- MKMapView determines when annotations views should be created
- MKMapView asks delegate for a MKAnnotationView for each MKAnnotation model object
- If no view is provided, a pin view is created
- MKMapView positions each view

Annotation Flow

Step 1

Application

MKMapView



App creates an annotation and adds it to MKMapView

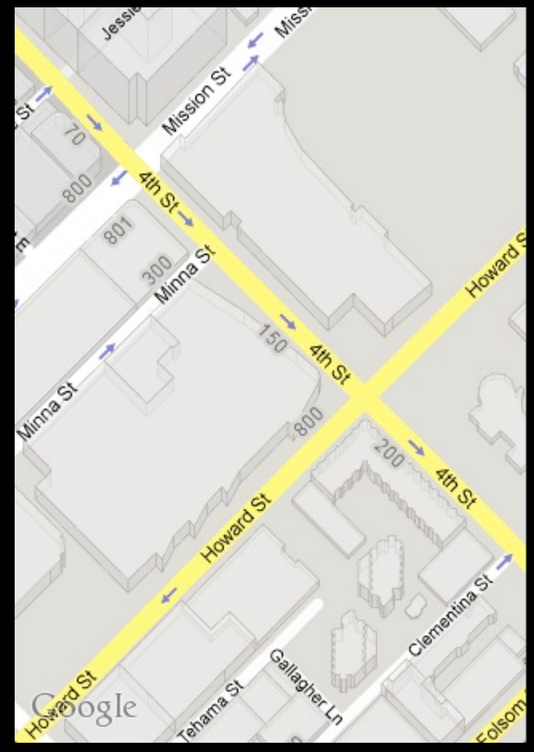
Annotation Flow

Step 2

Delegate

`mapView:viewForAnnotation`

MKMapView



MKMapView asks the delegate to create an annotation view

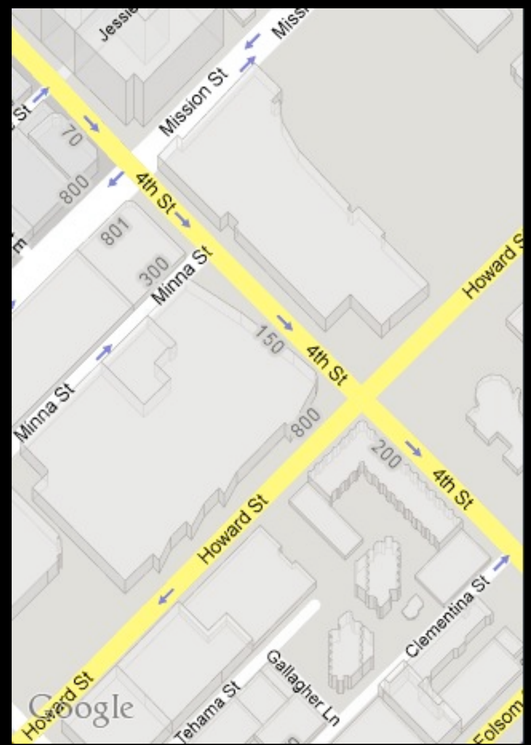
Annotation Flow

Step 3

Delegate

`mapView:viewForAnnotation`

MKMapView



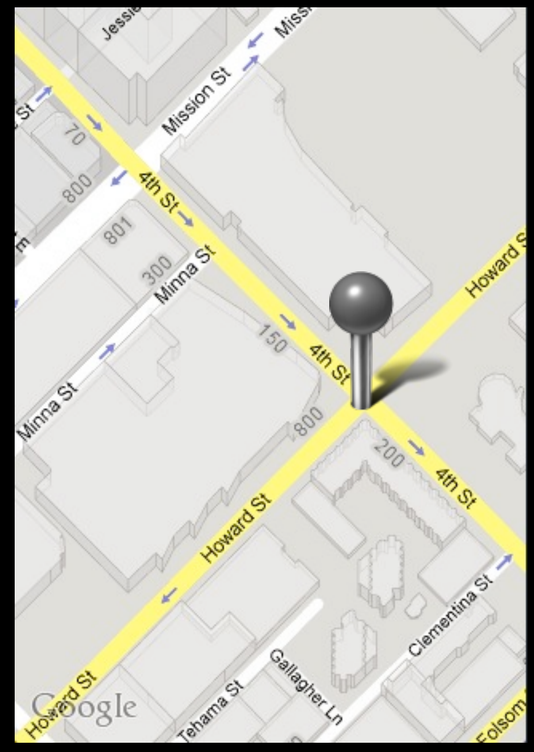
App creates an annotation view and returns it

Annotation Flow

Step 4

Delegate

MKMapView



MKMapView positions the annotation view

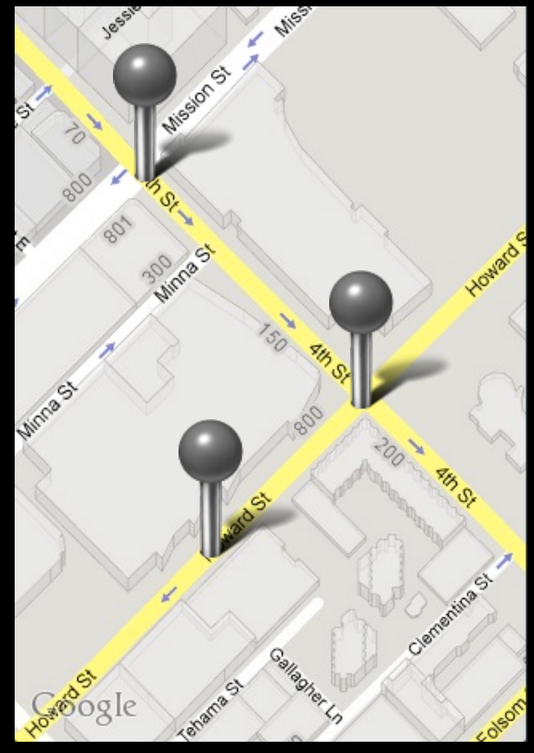
Annotation Flow

Step 5

Delegate

```
mapView:didAddAnnotationViews:
```

MKMapView



Delegate is told about all recently added annotation views

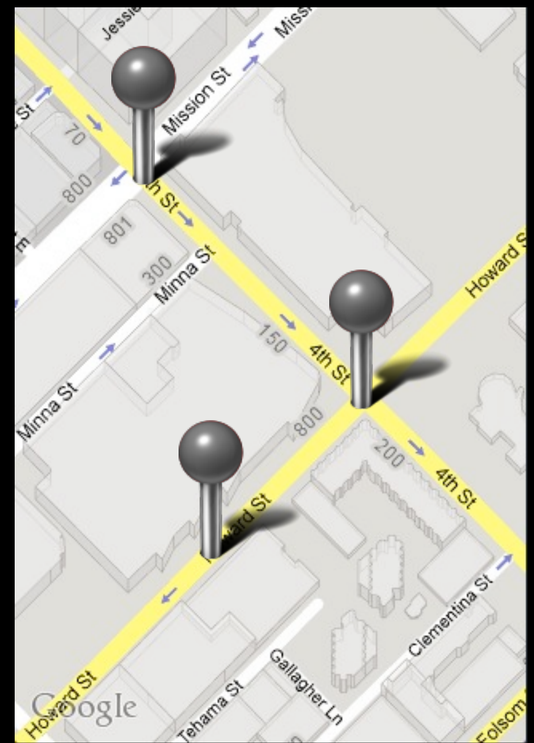
Annotation Flow

Step 6

Delegate

`mapView:didAddAnnotationViews:`

MKMapView



Delegate optionally animates added annotation views

Creating an Annotation View

Simple annotation with a custom icon

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView
            viewForAnnotation:(id <MKAnnotation>)annotation
{
  MKAnnotationView *view = [mapView dequeueReusableAnnotationViewWithIdentifier:@"logo"];
  if (view != nil)
    return view;

  view = [[MKAnnotationView alloc] initWithAnnotation:annotation reuseIdentifier:@"logo"];
  view.image = [UIImage imageNamed:@"logo.png"];
  return [view autorelease];
}
```

Creating an Annotation View

Animating Annotation Views

```
- (void)mapView:(MKMapView *)mapView didAddAnnotationViews:(NSArray *)views
{
    CGRect visibleRect = [mapView annotationVisibleRect];

    for (MKAnnotationView *view in views) {

        // Get the current frame, the destination of the view.
        CGRect endFrame = view.frame;

        // Place view offscreen
        CGRect startFrame = endFrame;
        startFrame.origin.y = visibleRect.origin.y - startFrame.size.height;
        view.frame = startFrame;

        // Animate view to its destination.
        [UIView beginAnimations:@”drop” context:NULL];
        view.frame = endFrame;
        [UIView commitAnimations];
    }
}
```

Creating an Annotation View

Annotation Positioning

- Annotation views are positioned based on the annotation's coordinate
- By default, the view's center is placed over the coordinate
- Use `centerOffset` to offset the view from the coordinate

Annotation Positioning

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView
    viewForAnnotation:(id <MKAnnotation>)annotation
{

...

    // Correctly position sign in the ground
    CGPoint centerOffset;
    centerOffset = CGPointMake(6, -48);

    view.centerOffset = centerOffset;

...

}
```

Annotation Selection

- Annotations are user selectable
- Selected programatically from MKMapView

Annotation Selection

- When selected, `selected` property on MKAnnotationView is YES
- Custom selection UI can be shown
- If `canShowCallout` property is YES, selection callout is shown

Annotation Callout

- Annotation `title` is required to show the callout
- `subtitle` is optionally shown below the `title`

Callout Accessories

- Callout supports left and right accessory views
- Accessory views are properties on MKAnnotationView

```
view.leftCalloutAccessoryView =
   [UIButton buttonWithType:UIButtonTypeInfoLight];

view.rightCalloutAccessoryView =
   [UIButton buttonWithType:UIButtonTypeDetailDisclosure];
```

Callout Accessories

- If accessories are UIControls, delegate is called when they are tapped
- No need to call `addTarget:action:forControlEvents:`

```
- (void)mapView:(MKMapView *)mapView
 annotationView:(MKAnnotationView *)view
calloutAccessoryControlTapped:(UIControl *)control
{
    [self showInfoAnnotation:view.annotation];
}
```

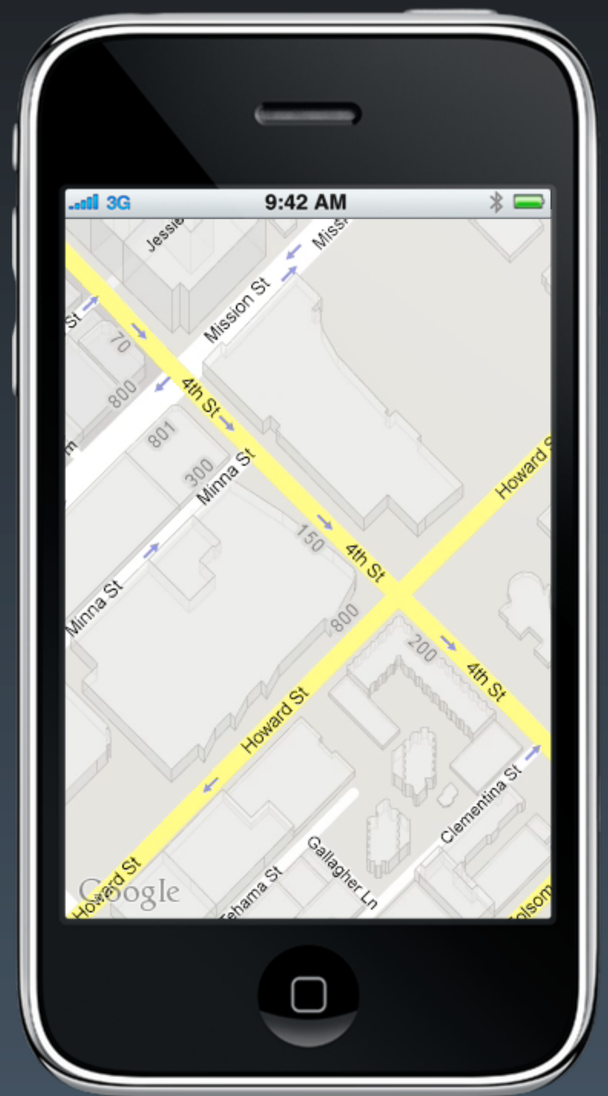
MKPinAnnotationView

- Subclass of MKAnnotationView
- Red, green or purple pins
  - Red is generic
  - Green for starting point
  - Purple for user-defined
- Pin drop animation is built-in using `animatesDrop` property

Demo

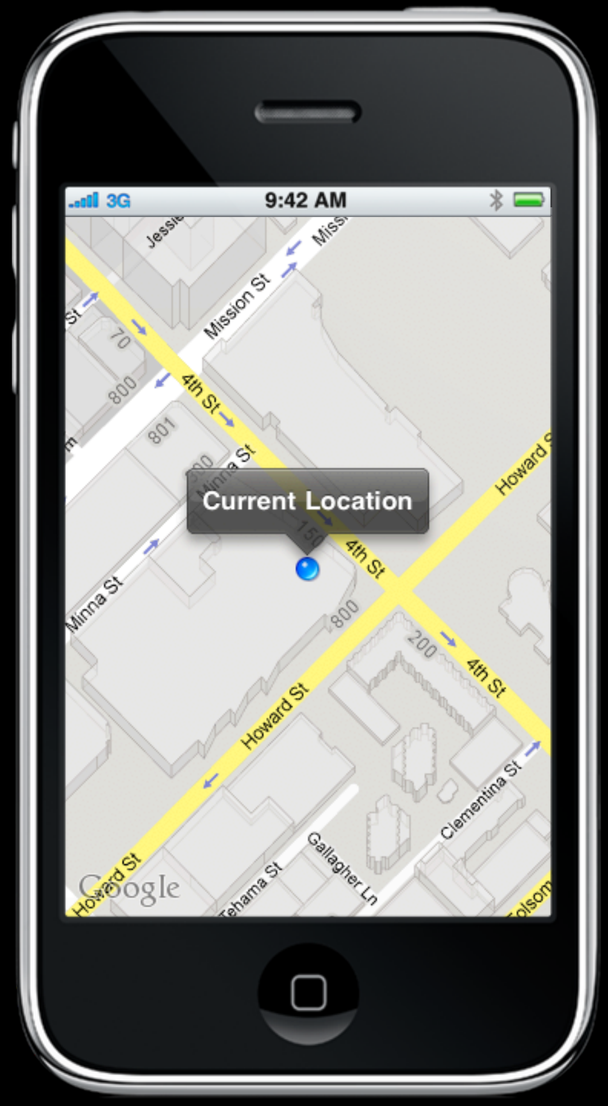
James Howard
Software Engineer

User Location



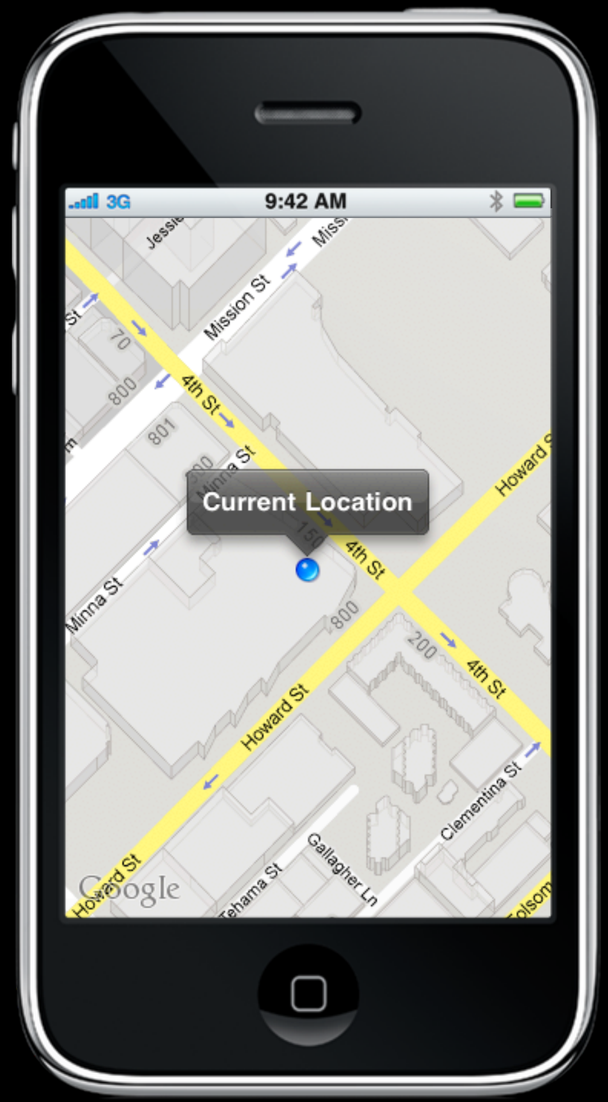
MKMapView User Location

- MKMapView uses CoreLocation
- `showsUserLocation` property shows user location once available
- User location is an annotation



MKUserLocation

- MKUserLocation is an annotation
- MKMapView provides the annotation view
- Delegate decides to use built-in view or custom view



Built-in User Location View

- Return nil to use built-in view

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView
            viewForAnnotation:(id <MKAnnotation>)annotation
{
    if (annotation == mapView.userLocation)
        return nil;

    ...
}
```

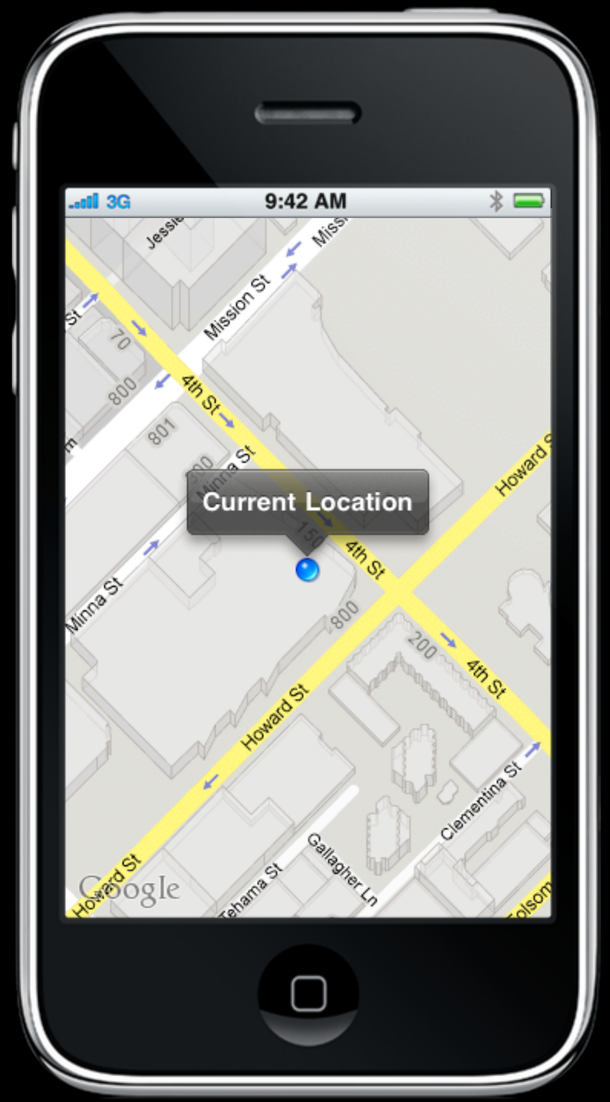
Custom User Location View

- Return your own user location view

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView
            viewForAnnotation:(id <MKAnnotation>)annotation
{
    if (annotation == mapView.userLocation) {
        MyLocationView *view;
        view = [[MyLocationView alloc] initWithAnnotation:annotation
                                           reuseIdentifier:nil];
        return [view autorelease];
    }

    ...
}
```

CoreLocation

- `location` property on MKUserLocation is a CLLocation object
- Different than location from CLLocationManager
- `location` is optimized for the map
- CLLocationManager provides the raw user location



Tracking Location Changes

- Use key-value observing to track user location changes

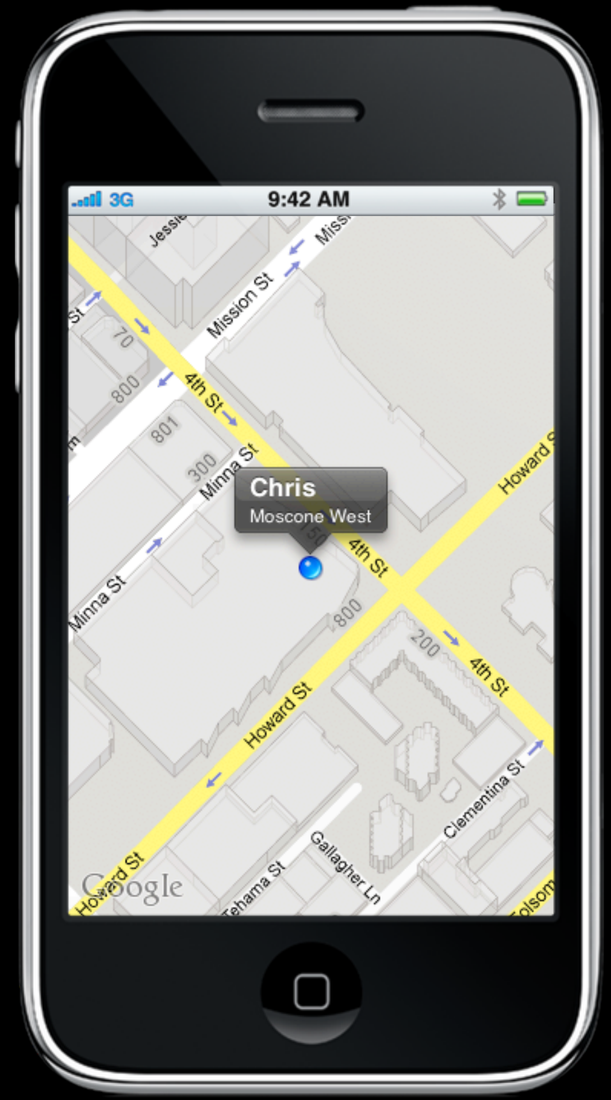
```
[mapView.userLocation addObserver:self
                       forKeyPath:@”location”
                          options:0
                          context:NULL];
```

```
- (void)observeValueForKeyPath:(NSString *)keyPath
                      ofObject:(id)object
                        change:(NSDictionary *)change
                       context:(void *)context
{
    // Go to user location.
    mapView.centerCoordinate = mapView.userLocation.location.coordinate;
}
```

MKUserLocation Title

- MKUserLocation `title` and `subtitle` are mutable

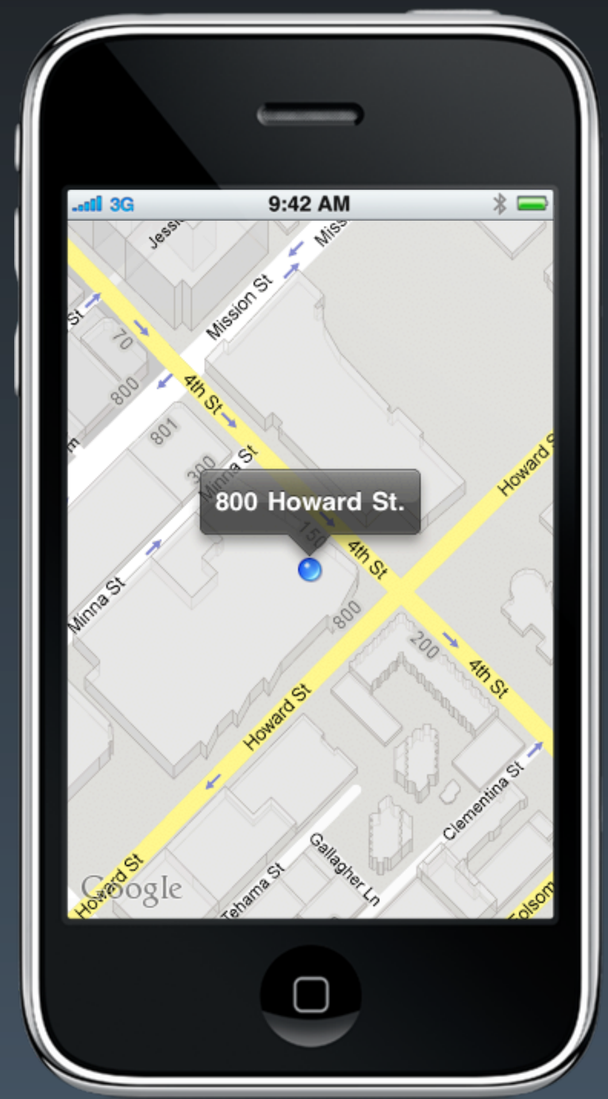
```
MKUserLocation *userLocation = mapView.userLocation;

userLocation.title = @”Chris”;
userLocation.subtitle = @”Moscone West”;
```



Demo

Kimon Tsinteris
Software Engineer

Geocoding


3G
9:42 AM
800 Howard St.
Mission St
4th St
Howard St
Minna St
Tehama St
Gallagher Ln
Clementina St
Google

Forward Geocoding

Turns an address into a coordinate

800 Howard St.
San Francisco, CA

37.78338, -122.403354

Reverse Geocoding

Turns a coordinate into an address

800 Howard St.
San Francisco, CA

37.78338, -122.403354

Reverse Geocoding

- MapKit only provides reverse geocoding
- Many geocoding services available online

MKReverseGeocoder

- Performs reverse geocoding queries
- Very useful when combined with CoreLocation
- Queries for placemark asynchronously
- Placemark is an address and its associated coordinate
- Delegate is called when query succeeds or fails

Reverse Geocoding

- Reverse geocode current location

```
CLLocationCoordinate2D coordinate = mapView.userLocation.coordinate;

MKReverseGeocoder *reverseGeocoder;
reverseGeocoder = [[MKReverseGeocoder alloc] initWithCoordinate:coordinate];
reverseGeocoder.delegate = self;

[reverseGeocoder start];
```

```
- (void)reverseGeocoder:(MKReverseGeocoder *)geocoder
       didFindPlacemark:(MKPlacemark *)placemark
{
    mapView.userLocation.subtitle = placemark.locality;
}
```

MKPlacemark

- The address for a coordinate
- Individual address fields are properties
- Dictionary representation can be used with Address Book
- It’s also an annotation

MKPlacemark

Property	Example
subThoroughfare	2000
thoroughfare	Mission St.
subLocality	Mission District
locality	San Francisco
subAdministrativeArea	San Francisco County
administrativeArea	CA
postalCode	94110
country	United States
countryCode	US

Demo

Seejo Pylappan
Software Engineer

Important Note

MapKit works with the Google Mobile Maps Service
http://code.google.com/apis/maps/iphone/terms.html

More Information

Matt Drance
Application Frameworks Evangelist
mdrance@apple.com

MapKit Documentation
http://developer.apple.com/iphone/

WWDC Attendee Site
http://developer.apple.com/wwdc/attendee/sessions/

Related Sessions

Finding Your Place with Core Location	Presidio Thursday 2:00PM

Labs

Core Location and MapKit Lab	iPhone Lab B Today 3:30PM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09